A Database-oriented Workflow Scheduler with Historical Data and Resource Substitution Possibilities

Tibor Dulai and Ágnes Werner-Stark
Department of Electrical Engineering and Information Systems, University of Pannonia, Egyetem str. 10, Veszprém, Hungary

Keywords: Database, Workflow Scheduler, Resource Substitution, Historical Data.

Abstract: This paper presents the database of a novel workflow scheduler that is able to handle resource substitution and takes into account historical data. The generated schedule can be optimized either in time or cost. The scheduler enables a resource substitution in case of an immediate event or when cost- or time-efficiency-related reasons necessitates it. The underlying database is able to handle complex workflows, represents the fleet of various resources and supports data mining from the data of the logged execution of the schedule in order to further improving the schedule. The database and the scheduler is a part of a complex project which schedules workflows described in XPDL by an agent system taking into account the real-time events and historical data served by process mining. Our scheduler system is intended to be applied both in business and industrial processes.

1 INTRODUCTION

Scheduling has huge importance for several reasons: companies are to increase their productivity with maintain their cost as low as possible. Nowadays the serious impact of some industrial processes is also realized, that's why the reduction of environment pollution is also an important goal of production planning. Suitable scheduling reduces the redundant use of polluting resources resulting greener processes. These factors motivated us in our work. We intend to develop a system which ensures effective scheduling capabilities both for business and industrial processes. Next to the usual functionalities of schedulers – e.g., resource allocations and optimization for time or cost domain – our system makes possible to take into account resource substitution possibilities an the experiences of previous executions. Resource substitution can take place in case of immediate events, e.g., when a resource breaks down and is not able to carry out its previously assigned tasks. The experiences from the previous executions are collected by the tools of process mining: historical data may improve the scheduling by using information e.g., about the trustworthiness or power of the resources in different circumstances.

For reaching our goals we had to create an appropriate data-model based database which has central role during the flexible and adaptive scheduling process. The database has to support the solution of wide-scale of scheduling problems where

- the processes to schedule have appropriate – in the database interactively predefined (via the user interface) – structure of jobs,
- each job may have arbitrary precondition(s), whose possible values have to be previously defined in the database,
- every resources of the problem have to be predefined in the database arbitrarily via the user interface,
- all the possible resource-job pairs with their properties (e.g. cost, operation time, resource operation mode, capacity, resource failure probability has to be previously defined in the database,
- all the entity types (e.g. process, job, resource type, resource) which can be related to a mined knowledge has to be previously defined in the database.

The goal of this paper is to introduce the structure of the database and the thoughts behind it, that made the creation of the adaptive scheduler possible.

Scheduling is widely investigated topic related to the execution of processes on computers or computer networks. Moreover, in case of business process planning and industrial production processes scheduler algorithms also have huge roles. Although, these algorithms are mainly applied before the processes starts, in offline manner. When circumstances change and an immediate event happens, the previously produced schedule is not usable or is not optimal any more. These cases necessitate real-time schedulers. Adaptive real-time schedulers are a relatively new topics of OR, see e.g., (Nandanwar and Shrawankar, 2012). Mainly these schedulers are constructed to apply them in multiprocessor computer systems (Rattanatamrong and Fortes, 2010), (Rattanatamrong and Fortes, 2011). We intend our system to respond to immediate events by taking into account the possible substitution of resources. Literature review showed that cooperation between the resources increases the effectiveness of the schedule, see e.g., (Murthy et al., 1997). Resource allocation and the whole scheduling can be improved when experiences from the past are used (Gregg et al., 2011). (Li, 2006) used data mining for collecting historical data about the behaviour of the schedule in the past. We acquire information about experiences of past executions also by applying process mining methods. (Ejarque, 2010) used historical data for predicate the behaviour of the tasks and resources in the system and for collecting information about the failures in the past. Next to these goals in our work we apply historical data for increasing the optimality of the prepared schedule.

A scheduler system is created for reaching all of these goals. We intend to include handling of real-time events, cooperation of the resources, adaptivity as well as mining and using of historical data into our interactive scheduler. The proposed system makes possible for the users to give different input data (workflow, resource properties, product types, etc.) via a graphical interface. It is capable to handle workflows in XPDL. Our sub-team gets these data in Json format and creates an agent system for dealing with them. As Figure 1 represents, there is a special agent for scheduling the resources which are represented by separate agent, there is an agent for mining data from the log entries of the previous executions, and there is an agent which deals with the database connection and operations.

The database of the system has a central role in supporting the process model representation, the scheduling and process mining functionalities in an easily usable way. In this paper this database is highlighted and the main functions are introduced based on its structure. Section 2 describes the structure and the intended content of the database. Based on the tables of the database and their relations Section 3 shows how we implement the two main function of the system: scheduling and process mining. Finally, Section 4 concludes our work.

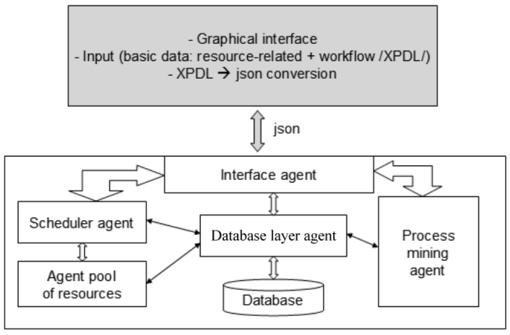

Figure 1: Overview of our scheduler system.

2 THE DATABASE OF THE SCHEDULER SYSTEM

The database of the system was created to make the resource-related data handling, the preparation of the schedule and process mining easy. The structure of the database is based on the formal model of the scheduler, which was presented in our former paper (Dulai et al., 2013).

The tables of the database can be clustered into three main groups:

- tables related to the workflow
- tables related to the resources
- tables related to process mining

These groups are introduced in the following subsections.

2.1 Workflow-related Tables of the Database

The workflow-related tables (see Figure 2) contain the workflow-model and the processes of the real executions, too, in the same structure. Each workflow belongs to a project, ensuring that users from different companies are able to use the system without mixing up their data. **Project** table contains the names of the different projects and its entries are linked to the data of the different users (companies).

The main goal of these table is to represent the structure of the actions in the processes. Important information of the processes – like e.g., its start and its latest possible finish time – are stored in the **Process** table. The elements of the processes are the actions, its main properties – e.g., its name, the real start time of the action, whether it is a pause, on which product type is it carried out, how many pieces of the product type are involved into this action, what is the maximal pause after executing the action which is tolerated, how the default operation time and cost are

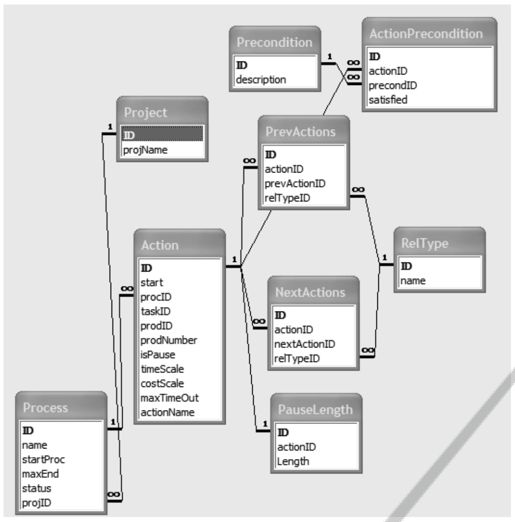

Figure 2: The workflow-related tables of the database.

scaled by the user – can be found in the **Action** table of the database. Actions of a process may have different relations with each others (e.g., an action may have 4 different direct following actions where all of them have to be executed or there is only one following action in direct sequential relation or there may be any other relation between the actions of a process). This flexibility has to be ensured by the data representation in the database, too. It is done by the **PrevActions** and **NextActions** tables which have to be handled together. These tables are for describing the neighbouring actions of a process, while a special table (**RelType**) represents the relation type of the neighbouring actions. This way the possible relation types can be defined arbitrarily.

There may exist a special case: an empty action, the pause. During pause there is no need for resource in the process execution, we only have to wait. The pause is represented by a special action, which has only one important property: its length, which is contained in the **PauseLength** table.

There are cases when actions may be executed only when some preconditions are satisfied. The database makes possible to define arbitrary preconditions in its **Precondition** table. These can be assigned to the actions of processes via creating entries in **ActionPrecondition** table. It makes possible to define actions without precondition, while there may exist different actions which have the same precondition.

2.2 Resource-related Tables of the Database

This section describes the part of the database which belongs to the concrete schedule and execution of the processes. When a process is scheduled, its model is concretized by assigning its actions to concrete resources with concrete parameters. The same can be stated for the case of executed processes, too. These tables for containing concrete resource-related and process execution related information can be seen in Figure 3.

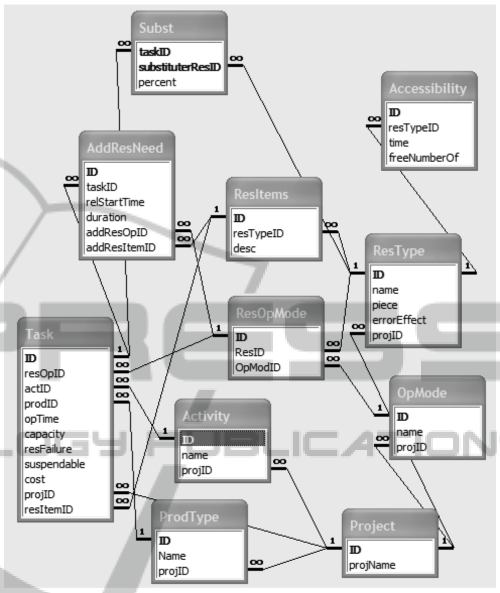

Figure 3: The resource-related tables of the database.

The system we created deals with resource types. Companies can determine how many pieces of the different resource types they have, what is their name and in case of failure how long are they unusable. These properties of a resource type can be found in the **ResType** table. Table **Accessibility** describes how many resource of a given resource type is available – in fact this table contains the timestamps when the availability of a resource changes and the cardinality of the set of the available resources of the referred type starting from the timestamp. In some cases resource type is not enough, we have to refer a concrete resource for information assignment (e.g., by applying process mining different efficiency can be determined for different resources of the same type). Table **ResItem** makes possible the identification and characterization of a concrete resource.

An important property of the resource types – which makes possible them to substitute other resource types – is that a resource type may have different operation modes. The database makes possible to define operation modes in **OpMode** table and to assign them to resources by writing **ResOpMode** table.

In our system the basic properties of the processing of an action by a resource is determined by four different factors:

- the resource (or the type of the resource)
- the applied operation mode of the resource
- the activity which is carried out by the resource
- the product type the action is applied on – its names are stored in table **ProdType**

This quartet determines the main properties of the concrete execution of an action, like the duration of the operation, its cost, its failure probability, and the capacity of the resource. These data are stored in the **Task** table. One more property belongs here, which is important in real-time scheduling: it indicates whether the action is suspendable during the execution without starting it from its beginning again. Users may name the concrete executed actions, this data is placed into table **Activity**.

Some actions require additional resources during the part (or the whole time) of their execution. These needs are stored in the **AddResNeed** table, where next to the ID of the additional resource and its operation mode the starting time and duration of the needed usage can also be set.

In the creation of optimal schedule table **Subst** has huge importance. This table stores the basic information about the substitution capability of the resource types. Of course different resource types may carry out the same action with different efficiency, e.g., resource B needs twice more time or cost for doing the same action on a product than resource A. This relation of the substitutable resource types is also stored in the table.

Similarly to the workflow-related data, resource-related basic data are also project-dependent, they are assigned to the entries of table **Project**.

2.3 Process Mining-Related Tables of the Database

Process mining algorithms are used for extracting important historical data from former process execution logs. Then, these data are used to refine the schedule. The database is prepared for supporting process mining and the storage of the mined information. Figure 4 shows the tables which are responsible for dealing with historical data.

Log table stores the event log entries. The main purpose of this block was to be general enough because of the wide usability. In this table the timestamp of the log entry, its description and status are stored, together with the identification of the related action. This action has further information about the circumstances of the formation of the log entry in the Task table whose relevant entry is referenced in the Action table.

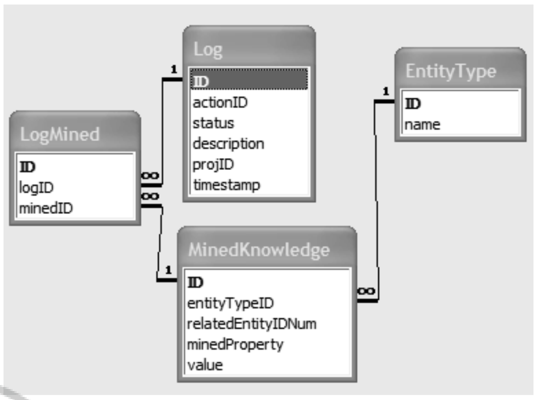

Figure 4: The process mining-related tables of the database.

After process mining algorithm processed the log entries, mined knowledge is issued. This knowledge is stored in the **MinedKnowledge** table. As the mined information can be varied, the structure of its storage has to be flexible. We store general property-value pairs in the table. Of course we should know which entity (action, resource, etc.) the mined knowledge refers. The type of the referred entity (which has to be previously declared in the **EntityType** table) and its ID are also stored together with the mined knowledge in the table. Finally, the origin of the mined knowledge (which log entry or log entries are they origin from) can be assigned in the **LogMined** table.

3 SCHEDULING AND PROCESS MINING FUNCTIONS OF THE SYSTEM

Based on our previously introduced database, in this section we briefly present the main functions of the scheduler system: how the scheduler algorithm works taking into account the possible resource substitution and how historical data are used for improving the schedule. Finally, we collect the advantages of our approach, these advantages origin from the structure of the database.

3.1 Basic Scheduling Method Built on Resource Substitution Possibilities

The initial state our scheduling system is count on that the basic data are written into the database. These data are the types of the resources which are at service at the company; the properties – like possible operation modes, cost, operation time, etc. – of these resource types; the activities which have to be carried out, etc.

After these data are entered via the GUI, the system can handle the workflows as follows:

- A model of the workflow has to be created by the user. It has to express the desirable operation of the workflow. It means the identification of the proper sequence of the actions of the workflow and the desires resource allocation. In this phase the user does not have to deal with the availability of the resources.
- Based on the workflow model and the basic data of the resources stored in the database, the scheduler allocates the proper resources to the proper actions of the workflow. In our case it is done following the goal that an action has to be started as soon as possible. If the resource which was assigned in the model is not available – it has to be verified in the Accessibility table of the database – , then the scheduler looks for a possible substitute based on the Subst table of the database. When there is not free resource to carry out the action, then the beginning of the action has to be shifted in time forward while a resource becomes free which can carry out the action. Following these steps for action to action, the final schedule will be prepared.
- When the real execution of the workflow takes place, the steps of the execution are written into the database the same way as in case of the workflow model. Next to the fixation of the real sequence of the actions and the resources which were involved into the execution, the logs of the actions are prepared, too, and are written into the Log table of the database. Of course, immediate events may result that the real execution differs from the schedule.
- Based on the time to time updated content of the log table, the process miner agent collects data which are written into the MinedKnowledge table of the database. These collected information can be used by the scheduler agent in later schedules and for improving recent schedules, too.

3.2 Usage of Historical Data for Improving Scheduling

The Process Mining agent collects experiences about the real executions of workflows which can be useful for later schedules. These experiences are information which characterize the different elements of the process execution: the resources, the actions, etc. Mined information are stored in the MinedKnowledge table of the database. We collect information for getting knowledge about:

- Which resource is the most time- or cost-effective in average in carrying out a given action?
- Which resources are overloaded?
- Does the direct previous action of a resource influence its time- or cost-efficiency during performing an action?
- Does the simultaneous operation of two resources influence their time- or cost-efficiency during performing actions?
- Does the absolute time of the work influence the time- or cost-efficiency of performing an action by a resource?
- How the failure-probability – or other important properties – of the resources change over time?
- Which action resulted the most failure?
- etc.

These information can be used by the scheduler agent in resource selection for carrying out an action. Historical data may be talkative about the expectable performance of the resources.

In the future we intend to implement more scheduler algorithms (e.g., heuristic schedulers) and compare their efficiency. Moreover, our goal is to collect several information of the past workflow executions and implementing more and more process mining algorithms for improving the final schedule, while verifying the impact of the application of the mined information on the efficiency.

3.3 Advantages of the Approach which origin of the Database Structure and Content

The database presented in Section 2 makes the reach of our goals possible: the inclusion of the wide spectra of improvements into our scheduler, related to both the reaction to the immediate dynamic events, the cooperation of the resources and the improvements helped by mined historical data. The structure and the content of the database ensure the following possibilities:

- The system is able to handle several separate projects parallel.
- the suspendability of the activities can be taken into account during the creation of the schedule. It may result that an ongoing action can be suspended to share its resource with an action which begins later without starting the former action again from its beginning.
- Possibility of resource failure can be taken into account during schedule preparation.

- Experiences of the ongoing execution can influence in real-time the parameters of the schedule.
- Different operation modes of the same resource can be taken into account.
- The substitutability of resources and its scale – the difference of different resources performing the same action – can be handled.
- Preconditions can be assigned to the actions. More than one action can have the same precondition, too.
- Historical data can be stored.
- The results of process mining can be identified, collected, their relation with their origin and subject can be expressed.

These advantages ensure that a flexible and up-to-date scheduling system can be created which meet with our exceptions.

4 CONCLUSIONS

This paper presented an ongoing research of a workflow scheduler which takes into account resource substitution possibilities and historical data which origin from past executions. This scheduler is a part of a complex system which is developed for scheduling business processes, however, we intend to apply it for other processes, too.

The paper places the database of the scheduler into the centre. This database stores properties of the resources, stores the model of the workflow and after scheduling its result is also written into the database. Moreover during the real execution, log entries and the possible changes are also fixed in the database. The paper details the structure of the database, and highlights the relation of the main functions of the scheduler and the database tables. For extracting useful information of historical data we implement process mining algorithms. We intend to analyze their effect on the efficiency of the scheduler. Moreover, implementing more scheduler algorithms we want to use our system for comparing them in the future.

ACKNOWLEDGEMENTS

This publication/research has been supported by the European Union and Hungary and co-financed by the European Social Fund through the project TAMOP-4.2.2.C-11/1/KONV-2012-0004 National Research Center for Development and Market Introduction of Advanced Information and Communication Technologies.

We acknowledge the financial support of the Hungarian State and the European Union under the TAMOP-4.2.2.A-11/1/KONV-2012-0072.

The authors wish to thank Prof. Katalin M. Hangos from Computer and Automation Research Institute, Budapest, Hungary, for her advices during the creation of this paper.

REFERENCES

Dulai, T., Werner-Stark, A., and Hangos, K. (2013). Immediate event-aware model and algorithm of a general scheduler. *Hungarian Journal of Industry and Chemistry*, 41(1):27–34.

Ejarque, Jorge; Micsik, A. S. R. P. P. K. L. B. R. M. (2010). Semantic resource allocation with historical data based predictions. *Proc. of Cloud Computing 2010 : The First International Conference on Cloud Computing, GRIDs, and Virtualization*, pages 104–109.

Gregg, C., Boyer, M., Hazelwood, K., and Skadron, K. (2011). Dynamic heterogeneous scheduling decisions using historical runtime data. *Proc. 2nd Workshop on Application for Multi- and Many-Core Processors*. San Jose, CA.

Li, X. (2006). Application of data mining in scheduling of single machine system. *PhD dissertation*. Iowa State University.

Murthy, S., Akkiraju, R., Rachlin, J., and Wu, F. (1997). Agent-based cooperative scheduling. *Proc. AAAI Workshop on Constraints and Agents*, pages 112–117.

Nandanwar, J. and Shrawankar, U. (2012). An adaptive real time task scheduler. *IJCSI International Journal of Computer Science Issues*, 9(6/1):335–340.

Rattanatamrong, P. and Fortes, J. A. B. (2010). Real-time scheduling of mixture-of-experts systems with limited resources. In Johansson, K. H. and Yi, W., editors, *HSCC*, pages 71–80. ACM.

Rattanatamrong, P. and Fortes, J. A. B. (2011). Real-time scheduling of ensemble systems with limited resources. *PhD dissertation*. University of Florida.